

Kernels in planar digraphs

Gregory Gutin*

Department of Computer Science
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK, gutin@cs.rhul.ac.uk

Ton Kloks[†]

Department of Mathematics and Computer Science
The University of Lethbridge, Alberta, T1K 3M4, Canada
kloks@cs.uleth.ca

Chuan Min Lee

Department of Computer Science and Information Engineering
Chung Cheng University, Ming-Shiun, Chaiyi 621, Taiwan
cmlee@cs.ccu.edu.tw

Anders Yeo[‡]

Department of Computer Science
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK, anders@cs.rhul.ac.uk

*Corresponding author. Research of Gutin was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778 and by a Leverhulme Trust grant

[†]Research of Kloks was supported by an EPSRC grant

[‡]Research of Yeo was supported in part by a Leverhulme Trust grant

Abstract

A set S of vertices in a digraph $D = (V, A)$ is a kernel if S is independent and every vertex in $V - S$ has an out-neighbor in S . We show that there exist $O(n2^{19.1\sqrt{k}} + n^4)$ -time and $O(k^{36} + 2^{19.1\sqrt{k}}k^9 + n^2)$ -time algorithms for checking whether a planar digraph D of order n has a kernel with at most k vertices. Moreover, if D has a kernel of size at most k , the algorithms find such a kernel of minimal size.

Keywords: Kernels; Planar Digraphs; Fixed-parameter complexity

1 Introduction

Let D be a digraph. For an arc xy in D , y is an *out-neighbor* of x and x is an *in-neighbor* of y . The number of out-neighbors (in-neighbors) of x is denoted by $d^+(x)$ ($d^-(x)$); $d(x) = d^+(x) + d^-(x)$. A set S of vertices is a *kernel* if S is independent and every vertex in $V(D) - S$ has an out-neighbor in S .

Notice that not every digraph has a kernel. For example, the directed 3-cycle (with vertices x, y, z and arcs $(x, y), (y, z), (z, y)$) has no kernel. In fact, all odd length directed cycles and most tournaments (orientations of complete graphs) have no kernels. It is easy to see that every acyclic digraph has a kernel. This sufficient condition for a digraph to have a kernel has been generalized by several authors. For short accounts on the topic see [4, 6].

Kernels in digraphs were introduced in different ways in [22, 27]. It seems that von Neumann and Morgenstern [27] were the first to introduce kernels when describing winning positions in 2-person games. For important applications of kernels in game theory see [13, 14, 26]. Applications of kernels are widespread and appear in diverse fields such as logic, computational complexity, artificial intelligence, graph theory, combinatorics and coding theory. For recent applications to counterexamples to the 0-1 laws in fragments of monadic second order logic, see, e.g., [24, 25].

Chvátal proved (see [18]) that the problem of deciding whether a digraph has a kernel is NP-complete. Fraenkel [13] showed that the problem remains NP-complete even for planar digraphs D with degree constraints $d^+(x) \leq 2$, $d^-(x) \leq 2$ and $d(x) \leq 3$ for all vertices x . Finding kernels in special classes of digraphs seems to be a mostly open field of study. It has been shown [5] that the kernel problem is polynomial time solvable for *locally semicomplete digraphs*, digraphs in which the out-neighbors (in-neighbors) of every vertex are adjacent. Kernels in some classes of planar graphs were investigated in [23].

In this paper we study the problem of finding a kernel in a digraph using methods of parameterized complexity [10]; see [10] for all undefined parameterized complexity terms.

Definition 1.1 *Consider an algorithm for a parameterized problem (I, k) , where I is the problem instance and k the parameter. The algorithm is called uniformly polynomial if it*

runs in time $O(f(k)|I|^c)$, where $|I|$ is the size of I , $f(k)$ an arbitrary function, and c a constant independent of k . A parameterized problem is fixed-parameter tractable (FPT) if it admits a uniformly polynomial algorithm.

We study the following parameterized problem (KERNEL): given a digraph D and a positive integer k , as a parameter, check whether D has a kernel with at most k vertices. Similarly, we can define the dominating set problem (DOMINATING SET) and the independent dominating set problem (INDEPENDENT DOMINATING SET); a set S in an undirected graph G is dominating if any vertex in $V(G) - S$ has a neighbor in S ; S is independent dominating if S is both independent and dominating.

Notice that every graph has an independent dominating set as every maximal independent set is independent dominating. This is in contrast to the non-existence of kernels in some digraphs.

Since an independent dominating set in an undirected graph G is a kernel in the digraph obtained from G by replacing every edge $\{x, y\}$ by the pair xy, yx of arcs and since INDEPENDENT DOMINATING SET is $W[2]$ -complete (see [10], p. 464), KERNEL is $W[2]$ -hard. This means that KERNEL is fixed-parameter intractable unless very unlikely collapses occur of parameterized complexity classes (see [10] for details). Thus, in this paper, we concentrate on KERNEL for planar digraphs, PLANAR KERNEL. We will see that this problem is FTP (although its recognition version is NP-complete) and will derive relatively fast exact algorithms for the problem.

This situation is similar, in a sense, to that for DOMINATING SET. Several researchers studied DOMINATING SET (and some of its variations including INDEPENDENT DOMINATING SET) for planar undirected graphs. Already in 1995 [12] claimed an algorithm of complexity $O(11^k n)$ for checking whether a planar graph on n vertices has a dominating set with at most k vertices. The analysis of the algorithm there turned out to be flawed. In [2], a similar analysis is used to correctly prove the existence an $O(8^k n)$ -algorithm. An important breakthrough was discovery of an algorithm with sub-exponential $f(k)$; such a result, $O(c^{\sqrt{k}} n)$ with $c \leq 4^{6\sqrt{34}}$, was first obtained by Alber et al. [1]. Recently, the upper bound on the constant c was improved to 2^{27} [19] and further to $2^{15.13}$ [17]. Also, [2], [16] and [17] obtained algorithms of complexity $O(k8^k + n^3)$, $O(2^{16.48\sqrt{k}} + n^3)$ and $O(k^4 + 2^{15.13\sqrt{k}} k + n^3)$, respectively. Notice that the last three 'additive FPT' algorithms are of complexity cubic in n .

The $O(c^{\sqrt{k}} n)$ complexity has a good chance to be optimal, in a sense. Indeed, Cai and Juedes [9] showed that there cannot be a $2^{o(\sqrt{k})} n^{O(1)}$ algorithm for PLANAR DOMINATING SET unless $3SAT \in DTIME(2^{o(n)})$, which is considered to be unlikely.

Returning to PLANAR KERNEL, recall that it is more general than INDEPENDENT DOMINATING SET. Nevertheless, in this paper we show that some results in [1] and [19] together with several results specific for PLANAR KERNEL can be used to get an

$O(n2^{19.1\sqrt{k}} + n^4)$ -time algorithm and an $O(n^2 + 2^{19.1\sqrt{k}}k^9 + k^{36})$ -time algorithm for finding a minimal size kernel in a planar digraph D of order n if D has a kernel with at most k vertices or report that D has no kernel with at most k vertices. This, in particular, implies the existence polynomial algorithms for determining small kernels (of size $O(\log^2 n)$) in planar digraphs.

Our results are also of interest because the parameterized complexity of problems in digraphs has generally proved to be surprisingly difficult to establish, and a number of basic problems are still unresolved, such as DIRECTED FEEDBACK VERTEX SET (open even for planar digraphs), and the problem of determining whether a digraph has a subgraph consisting of two sets of vertices A and B , each of size k , with an arc from every vertex of A to every vertex of B . The last problem was posed in [21], a paper on data-mining the internet to identify on-line communities. The problem seems to be an obvious and perhaps easy candidate for $W[1]$ -hardness, but in fact has resisted much effort [15].

2 Small kernels in planar digraphs

We start from a well-known definition of a tree decomposition of a graph.

Definition 2.1 *A tree decomposition of an undirected graph $G = (V, E)$ is a pair (T, \mathcal{S}) , where T is a tree and \mathcal{S} is a set of subsets of vertices of G , called bags. \mathcal{S} is in 1-1 correspondence with the nodes of the tree T such that the following conditions are satisfied:*

1. *Every vertex of G is contained in at least one bag,*
2. *Both end-vertices of every edge are contained in at least one bag,*
3. *For every vertex x of the graph, if x appears in bags S_i and S_j then it appears in every bag corresponding to the vertices which lie on the path in the tree T between the nodes i and j .*

Notice that we call vertices of T *nodes* to distinguish them from the vertices of G . The *width* of a tree decomposition (T, \mathcal{S}) is the maximum cardinality of a bag S_i minus one. The *treewidth* of G is the minimum width over all possible tree decompositions of G .

Computing treewidth of graphs in general is NP-complete, however the problem is FPT. Moreover there exists a linear time algorithm to check if a graph has bounded treewidth [7, 20].

Alber et al. [1] were the first to prove that a planar graph G of domination number k has treewidth $O(\sqrt{k})$. They actually showed that the treewidth of G is at most $6\sqrt{34}\sqrt{k}+8$. This result was improved to $16.5\sqrt{k}+50$ by Kanj and Perkovic [19]. The current best result

of this kind is due to Fomin and Thilikos [17] (using branch decomposition algorithms by Seymour and Thomas [29] and a transformation from a branch decomposition to a tree decomposition by Robertson and Seymour [28]):

Theorem 2.2 [17] *Let G be a planar graph with n vertices. There is an $O(n^4)$ -time algorithm that either constructs a tree decomposition of G with $O(n)$ nodes and of width at most $9.55\sqrt{k}$, or determines that G has no dominating set of size at most k .*

To facilitate our description below we make use of a *nice tree decomposition* (see, e.g., [20]). In a nice tree decomposition, we have a *binary rooted tree* T , i.e., T is a rooted tree such that every node has at most two children. The nodes of T are of four types:

- *An insert node i .* The node i in T has only one child j and there is a vertex $x \in V$ not in S_j such that $S_i = S_j \cup \{x\}$.
- *A forget node i .* The node i in T has only one child j and there is a vertex $x \in V$ not in S_i such that $S_j = S_i \cup \{x\}$.
- *A join node i* has two children p and q . The bags S_i, S_p and S_q are exactly the same.
- *A leaf node i* is simply a leaf of T .

It is not hard to transform a tree decomposition of G into a nice tree decomposition. In fact, the following holds.

Lemma 2.3 [20] *Given a tree decomposition of a graph G with n vertices that has width k and $O(n)$ nodes, we can find a nice tree decomposition of G that also has width k and $O(n)$ nodes in time $O(n)$.*

The *underlying graph* of a digraph $D = (V, A)$ is a graph $G = (W, E)$ such that $W = V$ and $\{x, y\} \in E$ if and only if either $xy \in A$ or $yx \in A$ (or both). Our algorithm below is based on the following simple observation.

Proposition 2.4 *If a planar digraph D has a kernel of size at most k , then its underlying graph G has a dominating set of cardinality at most k .*

The proof of the following result is similar, but has certain differences with the proof of the corresponding theorem in [1].

Theorem 2.5 *Let D be a digraph of order n . Let the underlying graph G of D have a nice tree decomposition with $O(n)$ nodes and of width at most t . Then, in $O(n4^t)$ time, we can either find a minimum size kernel in D or determine that D has no kernel.*

Proof: Let (T, S) be a nice tree decomposition of G . Let S_1, S_2, \dots, S_r be the bags of the tree decomposition (i.e. the nodes of T are $1, 2, \dots, r$). Let $root$ denote the root node of T . Recall that every vertex (and arc) in D lies in at least one of the bags.

Let Y_i denote the union of the bags S_j of the subtree of T with root node i . For every i , consider a partition (K_i, MC_i, DC_i) of S_i (the three sets of a partition are disjoint and every vertex of S_i is in one of the sets). A (K_i, MC_i, DC_i) -kernel is an independent set Q in D such that $K_i \subseteq Q \subseteq Y_i$, $(DC_i \cup MC_i) \cap Q = \emptyset$ and every vertex in $Y_i - DC_i$ either lies in Q or has an out-neighbor in Q ¹.

The vertices in DC_i may have an out-neighbor in Q , or not. Since $(DC_i \cup MC_i) \cap Q = \emptyset$, every vertex in MC_i has an out-neighbor in Q . We define $\kappa_i(K_i, MC_i, DC_i)$ as the minimal size of a (K_i, MC_i, DC_i) -kernel, if one exists. If it does not exist, then $\kappa_i(K_i, MC_i, DC_i) = \infty$.

If we can compute $\kappa_i(K_i, MC_i, DC_i)$ for all partitions (K_i, MC_i, DC_i) and all i , then

$$\mu = \min\{\kappa_{root}(K, S_{root} - K, \emptyset) : K \subseteq S_{root}\} \quad (1)$$

gives us the size of a minimal size kernel in D .

Let i be a node of T . We show how to compute all possible $\kappa_i(K_i, MC_i, DC_i)$ in $O(4^t)$ time. In fact we can also compute the actual minimum (K_i, MC_i, DC_i) -kernels, for all possible partitions (K_i, MC_i, DC_i) in $O(4^t)$ time, but we will leave the details of this to the reader. This will imply the desired complexity above as T has $O(n)$ vertices. We consider the cases when i is a leaf, i has one child and i has two children, separately. We assume that if i does have some children, then all κ_i 's are known for these children. We will for each step argue that we find the correct values.

Case 1. Assume i is a leaf. There are $O(3^{|S_i|})$ distinct partitions (K_i, MC_i, DC_i) , and we can easily find all of these in $O(|S_i|3^{|S_i|})$ time. For each partition (K_i, MC_i, DC_i) we can check whether K_i is an independent set and every vertex in MC_i has an out-neighbor in K_i in time $O(|S_i|^2)$. If the outcomes of both checks are positive, we have $\kappa_i(K_i, MC_i, DC_i) = |K_i|$. Otherwise, we have $\kappa_i(K_i, MC_i, DC_i) = \infty$. This gives us a time complexity of $O(|S_i|3^{|S_i|} + |S_i|^23^{|S_i|}) \subseteq O(4^{|S_i|}) \subseteq O(4^t)$ (recall that $|S_i| \leq t + 1$).

Case 2. Assume i has one child. Let j be the child of i in T . By the definition of a nice tree decomposition, S_j and S_i are identical, except for one vertex, say x , which lies in either S_i or S_j . We consider the following cases.

If $x \in K_i$, then if x is adjacent to a vertex in K_i , then $\kappa_i(K_i, MC_i, DC_i) = \infty$. Otherwise set $DC_j = DC_i \cup N^-(x)$, $MC_j = MC_i - N^-(x)$ and $K_j = K_i - x$. Clearly $\kappa_i(K_i, MC_i, DC_i) = 1 + \kappa_j(K_j, MC_j, DC_j)$ now holds.

If $x \in MC_i$ and x has no out-neighbor in K_i , then we have $\kappa_i(K_i, MC_i, DC_i) = \infty$.

¹ MC and DC stand for Must Cover and Don't Care if a vertex from the set has an out-neighbor in the kernel

If $x \in DC_i$ or $x \in MC_i$ and x has an out-neighbor in K_i , then we have $\kappa_i(K_i, MC_i, DC_i) = \kappa_j(K_i, MC_i - x, DC_i - x)$.

If $x \in S_j$, then we have the following:

$$\kappa_i(K_i, MC_i, DC_i) = \min\{\kappa_j(K_i \cup \{x\}, MC_i, DC_i), \kappa_j(K_i, MC_i \cup \{x\}, DC_i)\}.$$

As all the above cases can be considered in $O(|S_i|)$ time, we get the time complexity $O(|S_i|3^{|S_i|}) = O(4^t)$ for computing κ_i 's for all possible partitions.

Case 3. Assume i has two children. Let j and l be the two children, and recall that $S_i = S_j = S_l$. It is not difficult to see that $\kappa_i(K_i, MC_i, DC_i)$ is equal to the minimum value of $\kappa_j(K_i, W, MC_i \cup DC_i - W) + \kappa_l(K_i, MC_i - W, DC_i \cup W) - |K_i|$, over all $W \subseteq MC_i$. The above can be done in $O(2^{|MC_i|})$ time and there are $\binom{|S_i|}{m} 2^{|S_i|-m}$ partitions (K_i, MC_i, DC_i) with $|MC_i| = m$. Thus, we can compute κ_i 's for all possible partitions of S_i in time $O(\sum_{m=0}^{|S_i|} 2^m \binom{|S_i|}{m} 2^{|S_i|-m}) = O(4^t)$.

Since each $\kappa_i(K_i, MC_i, DC_i)$ is computed correctly above, we note that our algorithm will return the correct value of μ in (1). If we remember a minimum (K_i, MC_i, DC_i) -kernel for every possible i and partition (K_i, MC_i, DC_i) , then our algorithm can in fact return the minimal sized kernel, and not only its size. Certainly, if $\mu = \infty$, D has no kernel. \square

By Theorems 2.5, 2.2, Lemma 2.3 and Proposition 2.4, we obtain the following:

Theorem 2.6 *Let D be a planar digraph of order n . There is an $O(n2^{19.1\sqrt{k}} + n^4)$ -time algorithm that checks whether D has a kernel of size at most k . Moreover, the algorithm finds a minimum size kernel in D , if D has a kernel of size at most k .*

This theorem and Proposition 2.4 imply

Theorem 2.7 *Let $D = (V, A)$ be a planar digraph of order n . In polynomial time, one can check whether D has a kernel of size $O(\log^2 n)$ and, if D has such a kernel, then find one of minimal size.*

In Section 3 we need the following extension of Theorem 2.6, which can be proved similarly to Theorem 2.6 (in every partition, we have $R \subseteq K_i$, $K_i \cap B = \emptyset$, otherwise $\kappa_i(K_i, MC_i, DC_i)$ would be set to ∞).

Theorem 2.8 *Let $D = (V, A)$ be a planar digraph and let R and B be disjoint subsets of V . An (R, B) -kernel is a kernel K with $R \subseteq K$ and $B \cap K = \emptyset$. There is an $O(n2^{19.1\sqrt{k}} + n^4)$ -time algorithm for checking whether D has an (R, B) -kernel of size at most k . Moreover, the algorithm finds a minimum size (R, B) -kernel in D , if D has an (R, B) -kernel of size at most k .*

3 Additive FPT algorithms

We start by giving a short description of a general approach to obtain an additive FPT algorithm from a multiplicative one (see [8, 11]).

Proposition 3.1 *Assume we have an algorithm which runs in time $O(f(k)n)$ for a parameterized problem (on digraphs) with parameter k for some function $f(k)$. Then, alternatively, we can obtain an algorithm which runs in time $O(n^2 + h(k))$ for some function $h(k)$.*

Proof: First compute a table T of solutions for all digraphs with at most $f(k)$ vertices. (This table is referred to as the “advice” in [8]). Now consider a digraph with n vertices. If $n \leq f(k)$ the solution can be looked up in T . On the other hand, if $n > f(k)$, our multiplicative algorithm runs in $O(f(k)n) = O(n^2)$ time. \square

Remark 3.2 *Since our digraphs have no parallel arcs or loops, we have an upper bound of $2^{c^{2\sqrt{k}}}$ for the number of digraphs with $c^{\sqrt{k}}$ vertices. Hence, direct application of the proof of Lemma 3.1 gives an (impractical) algorithm with running time at least $\Omega(n^2 + 2^{c^{2\sqrt{k}}})$. Notice that we can obtain an algorithm that runs in time $O(n^\alpha + h'(k))$ for any $\alpha > 1$ at the cost of a blow-up of the function h' .*

In the remainder of this section we describe an algorithm which is only singly exponential in \sqrt{k} and quadratic in n . Let D be a planar digraph. Assuming that D has a kernel of cardinality at most k we show first that we can reduce D to a digraph D' of order $O(k^3)$, and a set \mathcal{S} of subsets of vertices of D' , such that the following holds. If there is a kernel of size at most k in D , then the size of a minimum kernel in D will have the same size as a minimum kernel in D' , which contains at least one vertex from each set in \mathcal{S} , and fulfills some additional properties, which we will describe below. Furthermore there will be at most $O(k^9)$ subsets in \mathcal{S} . We describe how to construct D' below.

We color some of the vertices *red* in the process. We color the vertices of D' red if they *must* be contained in any kernel of cardinality at most k of D for a reason described below. All red vertices remain vertices of D' . Some vertices are removed from D' , and other vertices remain uncolored. During the process of constructing D' we keep the following condition invariant:

D has a kernel K of size at most k if and only if K is a kernel in D' containing all red vertices and such that for every set $S \in \mathcal{S}$ at least one element is in K .

Initially $D' = D$, all vertices are uncolored and $\mathcal{S} = \emptyset$. Clearly the invariant is valid in this initial stage.

Lemma 3.3 *If two vertices x and y of D have at least $3k + 1$ common neighbors, then at least one of x and y must be in a kernel of cardinality at most k .*

Proof: Let C be the set of the common neighbors of x and y , and let K be a kernel not containing x and y . Then, by planarity of D , any vertex $u \in K$ can be the out-neighbor to at most two of the vertices in C , and thus we have $2|K| \geq |C \setminus K|$. Clearly, $|K| \geq |C \cap K|$, and hence $3|K| \geq |C \setminus K| + |C \cap K| = |C|$, implying that $|K| \geq \lceil \frac{|C|}{3} \rceil$, i.e. if $|C| \geq 3k + 1$, then $|K| \geq k + 1$. \square

Lemma 3.4 *The number of pairs $\{x, y\}$ of vertices in D that have at least 3 common neighbors is at most $2n$.*

Proof: Consider a plane embedding of D . Assume that x and y have three common neighbors u, v, w . The five vertices induce a subgraph G of D in which exactly one vertex (u, v or w) does not belong to the unbounded face of the embedding of G . We call this vertex the *central vertex* of G . It remains to observe that any vertex of D may be the central vertex of at most two induced subgraphs of D of order 5 that have a pair of vertices with three common neighbors. \square

Our algorithm consists of three stages. The **first** stage in our construction of D' is as follows. Let $D' = D$ and $\mathcal{S} = \emptyset$. For every pair $\{x, y\}$ of vertices we check whether x, y have at least $3k + 1$ common neighbors in D , and if they do, then delete the common neighbors from D' . For every deleted vertex z we make a set S_z consisting of the out-neighbors of z that remain in D' and we remove z from all existing sets thus far. (A vertex z occurs in a set S_u only if z is an out-neighbor of u and if u is no vertex of D' .) We add to \mathcal{S} all sets S_z that has been formed in the end of the first stage. Notice that the invariant remains valid after completion of the first stage.

We will now show that the above can be done in $O(n^2)$ time. Let x be an arbitrary vertex. Mark all vertices adjacent to x , and for each $y \in V(D)$ count how many marked vertices y is adjacent to. This can be done in $O(n)$ time, as there are $O(n)$ edges in a planar graph. If y has at least $3k + 1$ marked neighbors, then $\{x, y\}$ is one of the pairs we were looking for above. So by repeating the procedure for every $x \in V(D)$, we can find all pairs $\{x, y\}$, with the above property, in $O(n^2)$ time. By Lemma 3.4 (and $k \geq 1$) there are at most $O(n)$ such pairs, so for each of them we can delete their common neighborhood, and remember the sets S_z in $O(n)$ time, resulting in an overall time complexity of $O(n^2)$ (note that there will be at most n sets in \mathcal{S} , as each of them is an out-neighborhood of a vertex in D , and there are at most n vertices in D).

The next lemma shows that vertices of large degree must belong to the kernel.

Lemma 3.5 *Assume that every pair of vertices in D' have at most $3k$ neighbors in common. If $d_{D'}(x) \geq 3k^2 + k + 1$, then x must be in any kernel of cardinality at most k .*

Proof: Consider a kernel K with at most k vertices. Suppose that $d_{D'}(x) \geq 3k^2 + k + 1$ and $x \notin K$. Let N denote the set of neighbors of x . Since every vertex of $N \setminus K$ is adjacent to a vertex in K and no pair x, u , where $u \in K$ have more than $3k$ common neighbors, we have $3k|K| \geq |N \setminus K|$. Since $|N \setminus K| \geq |N| - |K|$, we obtain $3k|K| \geq |N| - |K|$. Hence, $|N| \leq (3k + 1)k$, a contradiction. \square

This lemma leads to the validity of the **second** stage in our construction of D' : In this step we color every vertex x of D' red if its degree is at least $3k^2 + k + 1$ in D' . The in-neighbors of x are deleted from D' (they have a neighbor in the kernel), and the remaining out-neighbors y of x are deleted from D' and the out-neighbors of each such y are made sets S_y in \mathcal{S} . Notice that the invariant remains valid by Lemma 3.5. This can easily be done in $O(n^2)$ time, as it takes $O(n)$ time to compute all the degrees in D' , and at most $O(n^2)$ time to construct the sets S_y , as at most n such sets will be constructed, with at most n vertices in each.

The second stage completes the construction of D' .

Lemma 3.6 *If D' has a kernel with at most k vertices then $|V(D')| \leq k(3k^2 + k + 1)$.*

Proof: Let K be a kernel in D' with at most k vertices. Let $x \in K$. If x is red, then it has no neighbors in D' . If x is not red then its degree is at most $3k^2 + k$ by the definition of a red vertex. Finally, every vertex in $D' - K$ must be a neighbor of a vertex in K . It follows that the number of vertices in D' is at most $k(3k^2 + k + 1)$. \square

In the **third** stage of our algorithm we delete multiple copies of the same set in \mathcal{S} , for all sets of size one or two. This is not difficult to do for all sets of size one in $O(n)$ time, and for all sets of size two in $O(n^2)$ time. Simply run through all sets in \mathcal{S} , and mark the sets of size one or two which exist in \mathcal{S} , and then run through all sets of size one and two to see which ones have been marked. This gives the desired bound as $|\mathcal{S}| \leq n$. Since every set in \mathcal{S} is a subset of the out-neighbors of some vertex in D , we note that we cannot have three distinct vertices belonging to three sets in \mathcal{S} , as this would imply the existence of a subgraph isomorphic to $K_{3,3}$ in the underlying graph of D . Therefore $|\mathcal{S}|$ is at most $|V(D')| + |V(D')|^2 + 2|V(D')|^3$.

Notice that also in this final stage the invariant is valid. We next show that the number of sets in \mathcal{S} is at most a polynomial in k .

Lemma 3.7 *After stage 3, there exist at most $O(k^9)$ sets in \mathcal{S} . These sets can be found in time $O(n^2)$.*

Proof: By Lemma 3.6 and the above bound of $|V(D')| + |V(D')|^2 + 2|V(D')|^3$ on $|\mathcal{S}|$, we obtain the lemma. The time complexity has been proved above. \square

Combining the result of Lemma 3.7 with that of Lemma 3.6 we obtain:

Theorem 3.8 *There exists an $O(k^{36} + 2^{19.1\sqrt{k}}k^9 + n^2)$ -time algorithm for checking whether a planar digraph D of order n has a kernel of size at most k , and finding a minimal size kernel, if D has a kernel of size at most k .*

Proof: Construct a digraph F from D' and \mathcal{S} as follows. The vertex set of F equals the vertex set of D' extended with one new vertex for each set from \mathcal{S} . Each such vertex is colored *black*. Its out-neighborhood equals the set, and its in-neighborhood is \emptyset . The digraph F is planar because F is a subgraph of D by the definition of \mathcal{S} . By the above arguments, F can be constructed in time $O(n^2)$.

Let B be this set of black vertices and let R be the set of red vertices in D' . Clearly, we have reduced the problem of existence of kernel of cardinality at most k in D to the problem of finding an (R, B) -kernel of size at most k in F . By Theorem 2.8 there exist an algorithm for the last problem that runs in time $O(2^{19.1\sqrt{k}}|V(F)| + |V(F)|^4)$. Thus, $|V(F)| = O(k^9)$ implies the time complexity bound above. \square

Remark 3.9 *In fact, most of the above steps can be done in $O(n)$ time. However, it is unclear if it is possible to determine all pairs $\{x, y\}$, which have at least $3k + 1$ common neighbors in less than $\Theta(n^2)$ time. Even though we can do all other steps above in $O(n)$ time, we have chosen to give the $O(n^2)$ bounds, as this makes the proofs somewhat simpler.*

4 Discussion

In this paper, we applied fixed-parameter complexity approaches to develop relatively fast algorithms for finding minimal size kernels in planar digraphs that have kernels of size at most k . In particular, we obtained an $O(k^{36} + 2^{19.1\sqrt{k}}k^9 + n^2)$ -time algorithm. Since $f(k) = k^{36} + 2^{19.1\sqrt{k}}k^9$ is a fast growing function, the algorithm seems to lose practicality, in the worst case, even for relatively small values of k (see p. 13 of [10]). However, it may well be that our theoretical estimate of the worst case complexity is, in fact, far from optimal. Moreover, some preprocessing may improve efficiency of the algorithm.

Perhaps, better reductions will lead to faster algorithms for PLANAR KERNEL. In particular, it would be interesting to know whether in PLANAR KERNEL D can be reduced to F such that D has a kernel of size at most k if and only if F has a kernel of size k' , where the order of F is linear in k and $k' \leq ck$ for some constant c . (Unlike in our reduction above, no set \mathcal{S} is allowed.) For PLANAR DOMINATING SET such a reduction of both theoretical and practical significance was given in [3].

References

- [1] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for Dominating Set and related problems on planar graphs. *Algorithmica* 33 (2002), 461–493.
- [2] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond and U. Stege. A refined search tree technique for Dominating Set on planar graphs. To appear in *Journal of Computer and System Sciences*.
- [3] J. Alber, M. R. Fellows, R. Niedermeier. Polynomial time data reduction for Dominating Set. To appear in *Journal of the ACM*.
- [4] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, London, 2000.
- [5] J. Bang-Jensen, Y. Guo, G. Gutin, and L. Volkmann, A classification of locally semicomplete digraphs, *Discrete Mathematics* 167/168 (1997), 101–114.
- [6] C. Berge and P. Duchet, Recent problems and results about kernels in directed graphs, *Discrete Mathematics* 86 (1990), 27–31.
- [7] H. Bodlaender, A linear time algorithm for finding tree decompositions of small treewidth, *SIAM Journal on Computing* 25 (1996), 1305–1317.
- [8] L. Cai, J. Chen, R. Downey and M. Fellows. Advice classes of parameterised tractability, *Annals of Pure and Applied Logic* 84 (1997), 119–138.
- [9] L. Cai and D. W. Juedes, On the existence of subexponential-time parameterized algorithms. *Journal of Computer and System Sciences* 67 (2003), 789–807.
- [10] R.G. Downey and M.R. Fellows, *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag, 1999.
- [11] R.G. Downey, M.R. Fellows, and U. Stege, Parameterized complexity: A framework for systematically confronting computational intractability, *Contemporary trends in Discrete Mathematics*, (R. Graham, J. Kratochvíl, J. Nešetřil, and F. Roberts, eds.), Proceedings of the DIMACS–DIMATIA Workshop, Prague, 1997, *AMS–DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **49**, (1999) 49–99.
- [12] R.G. Downey and M. R. Fellows, Parameterized computational feasibility, In *Proc. of Feasible Mathematics II* (eds. P. Clote and J. B. Remmel), Birkhauser (1995), 219–244.
- [13] A.S. Fraenkel, Planar kernel and Grundy with $d \leq 3$, $d^+ \leq 2$, $d^- \leq 2$ are NP-complete, *Discrete Applied Mathematics* 3 (1981), 257–262.
- [14] A.S. Fraenkel, Combinatorial game theory foundations applied to digraph kernels, *Electronic Journal Combinatorics* 4 (1997), 17 pp.
- [15] M. Fellows, Private communication, 2001.
- [16] H. Fernau and D. Juedes, A geometric approach to parameterized algorithms for domination problems on planar graphs. *Proc. 29th International Symposium on Mathematical Foundations of Computer Science*, LNCS 3153 (2004), 488 – 499.

- [17] F.V. Fomin and D.T. Thilikos, Dominating sets in planar graphs: branch-width and exponential speed-up. In *Proc. 14th ACM-SIAM SODA* (2003), 168–177.
- [18] M.R. Garey and D. S. Johnson, *Computers and Intractability*. W. H. Freeman and Co., San Francisco, 1979.
- [19] I.A. Kanj and L. Perkovic, Improved parametrized algorithms for planar dominating set. In *Proc. 27th MFCS 2002*, Springer-Verlag LNCS 2420 (2002), 399–410.
- [20] T. Kloks, *Treewidth–Computations and approximations*, Springer-Verlag LNCS 842, 1994.
- [21] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A.S. Tomkins and E. Upfal. The Web as a graph. In *Proc. 19th ACM SIGACT-SIGMOD-AIGART Symp. Principles of Database Systems, PODS* (2000), 1–10.
- [22] M. Kwaśnik, The generalisation of Richardson theorem, *Discussiones Math. IV* (1981), 11–14.
- [23] C. Banderier, J-M. Le Bars and V. Ravelomana, Generating functions for kernels of digraphs. *Proc. 16th Annual International Conference on Formal Power Series and Algebraic Combinatorics* (2004), 91–105.
- [24] J-M. Le Bars, Counterexamples of the 0-1 law for fragments of existential second-order logic: an overview. *Bulletin of Symbolic Logic* 9 (2000), 67–82.
- [25] J-M. Le Bars, The 0-1 law fails for frame satisfiability of propositional modal logic. In *Proc. 17th Symposium on Logic in Computer Science*, 2002, 225–234.
- [26] J. van Leeuwen, Having a grundy numbering is NP-complete, Report 207, Computer Science Department, Pennsylvania State University, University Park, PA. 1976.
- [27] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, 1944.
- [28] N. Robertson and P.D. Seymour, Graph Minors. X. Obstructions to tree-decomposition. *J. Combin. Theory. Ser. B* 52 (1991) 153–190.
- [29] P.D. Seymour and R. Thomas, Call routing and the ratcatcher, *Combinatorica* 14 (1994), 217–241.